Perihelion Precession of Mercury's Orbit

Summer 2022 Project Report

Abeer Karandikar, Dhruv Pisharody

Mentored by

Professor Bikram Phookun
Philip Cherian

Department of Physics
Ashoka University
Sonipat, Haryana, India

Submitted on
September 25, 2022

Abstract

In pure Newtonian gravity, a small planet orbiting a much more massive star would revolve in a perfectly elliptical orbit. However, adding a correction term to the force law in corresponding with the effect of general relativity, causes this orbit to precess. In this project, we computationally simulate the orbit of Mercury around the Sun, and find the locations of the perihelion in each orbit. By taking the coordinates of each position of Mercury in polar coordinates, then using the deviation in the angular coordinate at the perihelia, we calculate the precession rate. To obtain more accurate results, we modified the force law in a manner that was linearly proportional to the precession rate, then extrapolated down to the 'true' force law.

Contents

Introduction

Before Albert Einstein brought forth his general theory of relativity, astronomers were baffled by the discrepancies they noticed between their theorized motion of Mercury around the Sun, and what they actually observed in the heavens. Mercury's orbit seemed to precess an extra 43 arcseconds per century, which Newtonian gravity, the primary theory used by scientists, could not explain. Though this might seem like a minuscule correction, the problem of the precession of Mercury's orbit became a significant scientific problem, eventually allowing for the verification of a new, ground-breaking theory of the universe.

In this project, we aim to model the Sun-Mercury system, both in pure Newtonian gravity, and with the effect of the correction from general relativity to the gravitational potential, and in each case reproduce the corresponding angular precession (or lack thereof) of the perihelion of Mercury.

Model

Background Calculations

The first step in solving the problem at hand was solving the two body problem, i.e. modelling the behaviour of any two bodies (in our scenario, the Sun and Mercury) interacting gravitationally. The problem can be simplified by assuming the Sun to be kept stationary at the origin and effectively converting it into an one-body problem. To solve this problem we can use step integration using the equations:

$$x(t + \delta t) = x(t) + \delta x = x(t) + v(t)\delta t$$
$$v(t + \delta t) = v(t) + \delta v = v(t) + a(t)\delta t$$

where $v = \frac{dx}{dt} = \dot{x}$ is the velocity, and $a = \frac{d^2x}{dt^2} = \ddot{x} = \dot{v}$ is the acceleration.

By dividing the trajectory of the body into n steps, the position x of the body at any step can be found out using the velocity v and acceleration a. The acceleration is given according to Newtonian gravity as,

$$\ddot{\mathbf{r}} = -\frac{\mathbf{F}(\mathbf{r})}{m} = -\frac{\mathbf{\Phi}(\mathbf{r})}{m} = -\frac{GM}{r^3}\mathbf{r} \tag{1}$$

where

$$G = 6.675 \times 10^{-11}\ \mathrm{Nm^2kg^{-2}} \text{ is the universal gravitational constant,}$$

$$M = 2 \times 10^{30}\ \mathrm{kg} \text{ is the mass of the Sun,}$$

$$m = 3.3 \times 10^{23}\ \mathrm{kg} \text{ is the mass of Mercury, and}$$

$$\mathbf{\Phi}(\mathbf{r}) = \frac{GMm}{r^2}\mathbf{r} \text{ is the gravitational potential in Newtonian gravity.}$$

Writing the vector $\mathbf{r}$ in polar coordinates (r, θ) and simplifying the resulting equations, we get two second order differential equations for the radial and angular acceleration:

$$\ddot{r} = -\frac{GM}{r^2} + \frac{l^2}{r^3} \tag{2}$$

$$\ddot{\theta} = -\frac{2\dot{r}\dot{\theta}}{r} = -\frac{2l\dot{r}}{r^3} \tag{3}$$

where $l = r^2\dot{\theta}$ is the angular momentum per unit mass of the second, orbiting body (Mercury).

The most basic algorithm for step integration is the Euler method, but better results can be obtained with different integration algorithms like the velocity Verlet and the Runge-Kutta-4 methods. To check which methods gives the best results, we decided to check conservation of energy in the orbits obtained from each method. The algorithm that conserves energy the best is the one we use for the rest of the project.

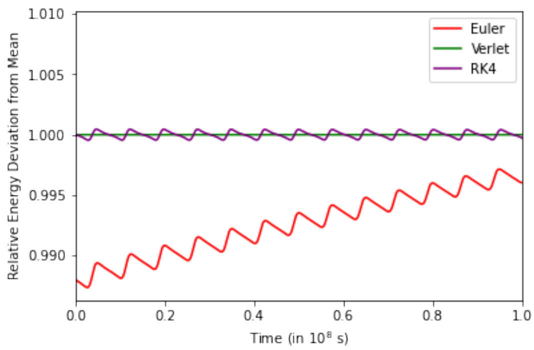

Figure 1: Energy divided by Mean Value, against Time, for Euler, Verlet, and RK4

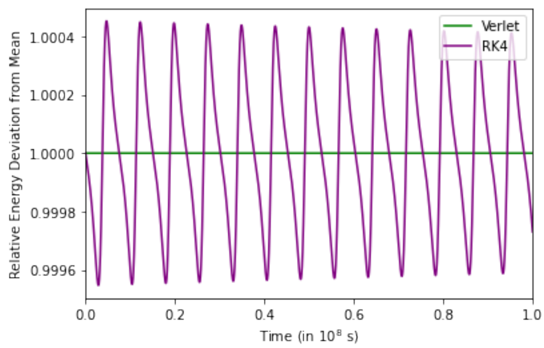

Figure 2: Energy divided by Mean Value, against Time, for Euler and Verlet

As the velocity Verlet algorithm shows the least deviation in the energy at each time step (among the three algorithms tried), it is thus the one we used for the rest of the project.

Here, instead of calculating the position, velocity, and acceleration at time steps $t_1, t_2, t_3, \ldots$, we calculate them at every half step $t_{0.5}, t_1, t_{1.5}, \ldots$, which results in the new equations for updating position and velocity,

$$x(t+\delta t) = x(t) + v(t)\delta t + \frac{1}{2}a(t)\delta t^2$$

$$v(t+\delta t) = v(t) + \frac{1}{2}(a(t-\delta t) + a(t))\delta t$$

Since we simulated our orbits in polar coordinates, the velocity Verlet integration is done simultaneously for both the radial position and velocity, and the angular position and velocity of Mercury.

Base Problem

Before dealing with the problem of general relativity, we must first simulate Mercury's orbit in base Newtonian gravity. Using equations (2) and (3), a non-precessing (or at least, as will be shown and explained later, an extremely mildly precessing) orbit is simulated.

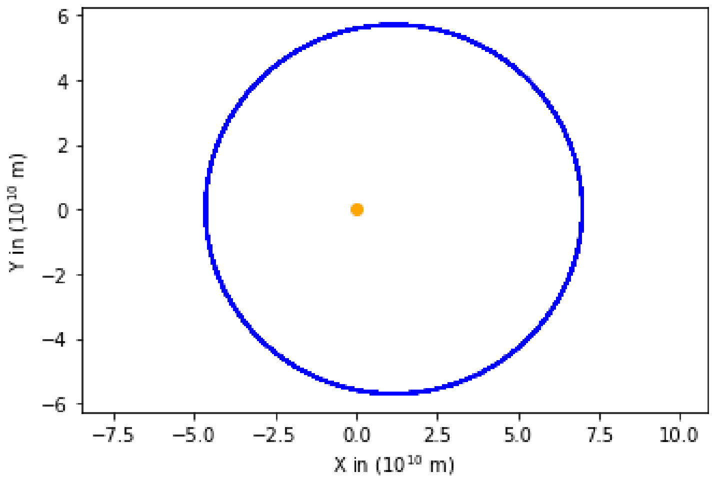

Figure 3: Mercury's orbit around the Sun in Newtonian gravity for 60 Mercury years

Adding the effect of General Relativity

The introduction of General Relativity requires adding an additional term to the gravitational potential expression. Now,

$$\mathbf{\Phi}(\mathbf{r}) = \left(\frac{GMm}{r^2} + \frac{3GMml^2}{r^5c^2}\right)\mathbf{r}$$

which makes the function for radial acceleration now

$$\ddot{r} = -\frac{\mathbf{F}(\mathbf{r})}{m} = -\frac{\mathbf{\Phi}(\mathbf{r})}{m} = -\frac{GM}{r^2} + \frac{l^2}{r^3} - \frac{3GMl^2}{r^4c^2} \tag{4}$$

Though the orbit looks much the same over the (cosmically relatively) short time scales we used, the perihelion is in fact precessing more rapidly.

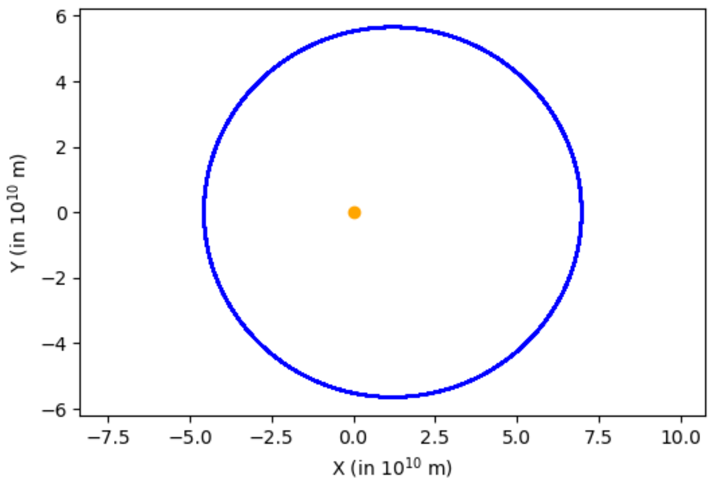

Figure 4: Mercury's orbit around the Sun with General Relativity for 60 years

Generalising the Acceleration Function

For more visible precession, we can modify the equation for acceleration [Equation (4)] to

$$\ddot{r} = -\frac{GM}{r^2} + \frac{l^2}{r^3} - \frac{kGMl^2}{r^4c^2} \tag{5}$$

In addition, as will be shown and used ahead, the variable k, which is 0 for Newtonian Gravity and 3 for General Relativity, is directly proportional to the precession rate of the perihelion.

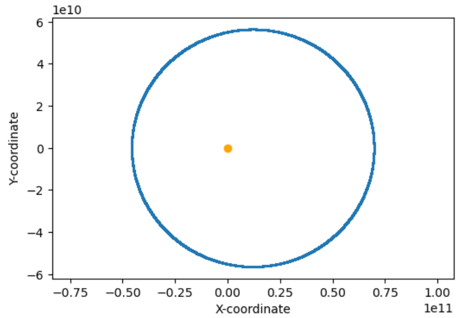

Figure 5: Mercury's orbit for $k = 3$

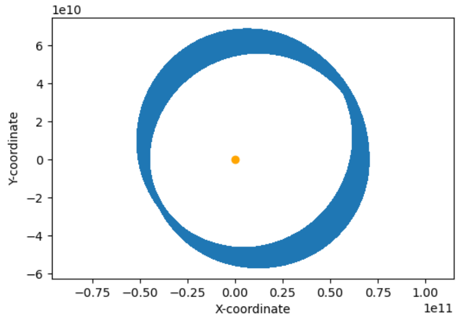

Figure 6: Mercury's orbit for $k = 50000$

Methods

Finding the Perihelia

To pick out the values of the distance between the Sun and Mercury at the perihelion, we compare the value of r at each time step with the values of r one step before and after it, and pick those values r_n which satisfy the condition,

$$r_n < r_{n+1} \text{ and } r_n < r_{n-1}$$

For each perihelion r_n, the corresponding θ_n and t_n can be obtained. Practically, this was performed by using the `find_peaks` function of the `scipy` Python package.

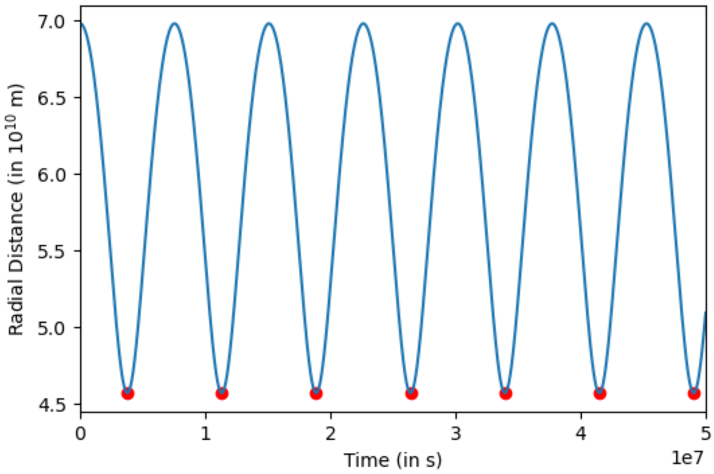

Figure 7: Position against Time, with the perihelia in red

Precession Rate

Once the angular coordinates θ of every perihelion have been picked out, the precession rate can be calculated. In a non-precessing orbit we expect to see that the values of θ at the perihelion are separated by exactly 2π from one to the next. In a precessing orbit, that is not the case and the values of θ at the perihelion have a difference slightly higher than 2π. By subtracting the appropriate multiple of 2π (or by taking θ modulo 2π), the deviation of the difference from 2π can be found. Since the same deviation is added at each perihelion, plotting the values of this deviation versus time should give a linear trend, the slope of which gives the precession rate.

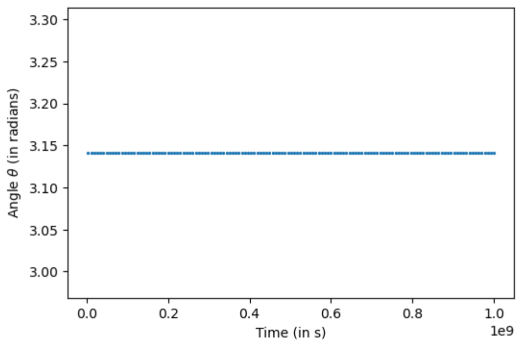

Figure 8: Expected Graph for Angle against Time for Newtonian Gravity

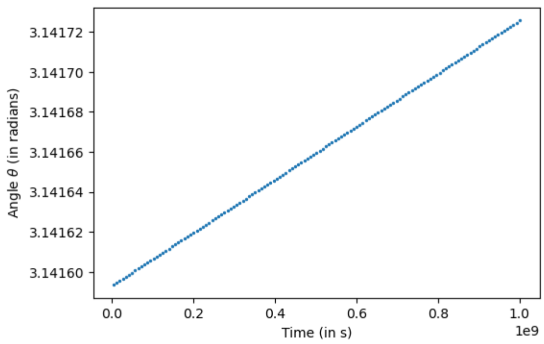

Figure 9: Expected Graph for Angle against Time for General Relativity

The difficulty in doing the process described above is that for small values of k (including the realistic value, $k = 3$), the angular deviation is extremely small. For these values, though the linear trend remains, numerical errors start to creep in.

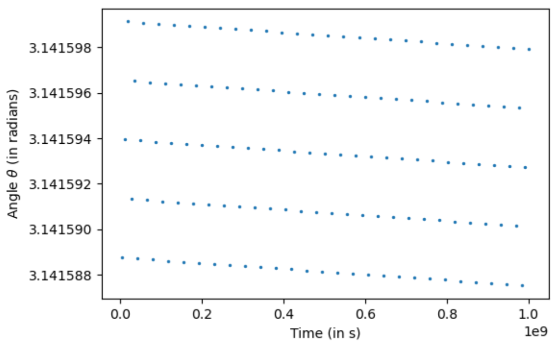

Figure 10: Angle against Time for Newtonian Gravity

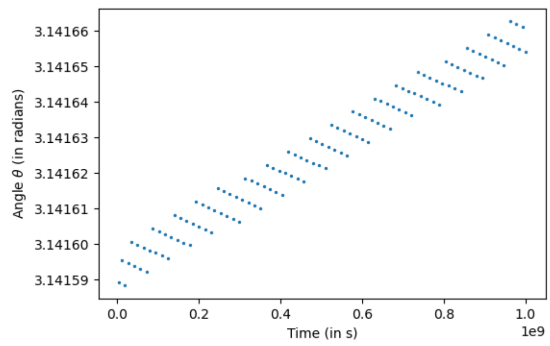

Figure 11: Angle against Time for General Relativity

The two possible ways to reduce these numerical errors are:

- decreasing the time step, or
- increasing the value of k to get more pronounced precession

The former quickly becomes unfeasible because with a hard minimum on the total time that must be simulated (at least one orbit), reducing the time step dramatically increases the computational time. Thus, though we went as far as we could with this possible solution (using a time step of 10 seconds), we resorted to the latter method to achieve more accurate results.

First, by increasing the value of k, we observed that higher values gave θ v/s t graphs that were more in line with what we theoretically predicted. The graphs became more linear even for values of k as low as 10.

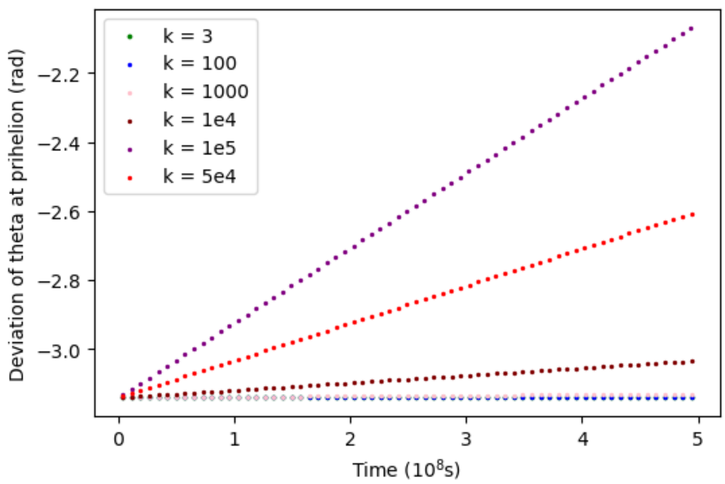

Figure 12: Angle against Time of the perihelia for various values of k

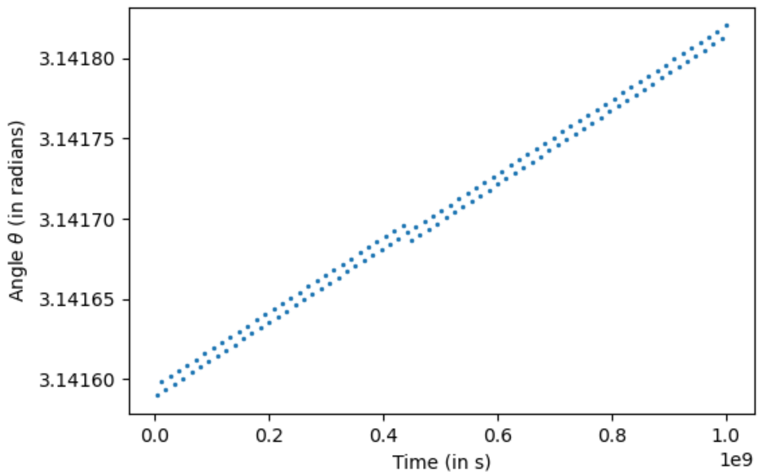

Figure 13: Angle against Time of the perihelia for $k = 10$

Additionally, by calculating the precession rate for the orbits produced by varying values of k, we observed that the precession rate was linearly proportional to the corresponding value of k.

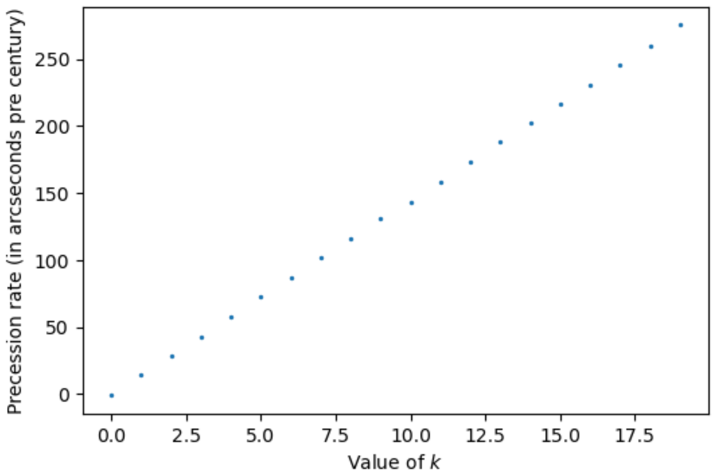

Figure 14: Linear Proportionality of k with Precession Rate

Extrapolating the values of angular precession for values of k going from 10 to 19, by dividing each value by the corresponding k then multiplying by 3, gives us

Value of k	Precession Rate extrapolated to $k = 3$ (in arcseconds per century)
10	42.93283
11	43.23908
12	43.28555
13	43.39118
14	43.42305
15	43.32551
16	43.28176
17	43.37463
18	43.36678
19	43.48188

Error Analysis

The simulation was run for 10^9 seconds with a time step (δt) of 10 seconds. The Verlet algorithm has an error factor of $(\delta t)^2$. Therefore the simulation gives accurate results till seven significant figures. However, rounding the values of θ give similar results as not rounding them.

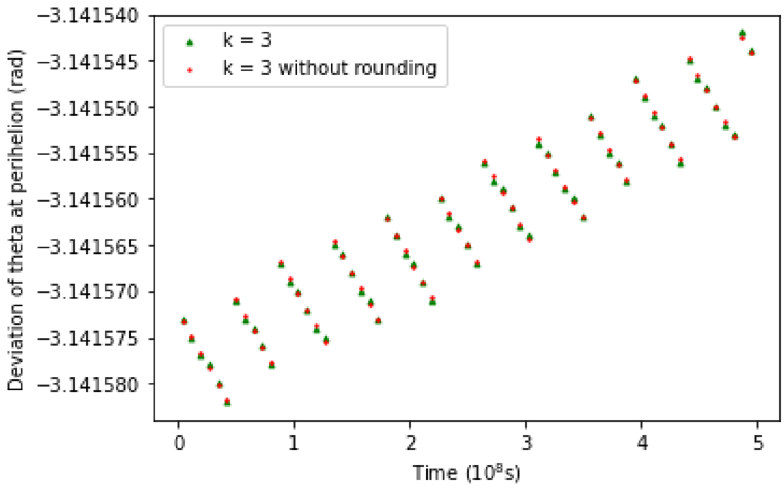

Figure 15: Deviation of θ for $k = 3$ With and Without Rounding

Results

Using graph 10 to calculate the rate of precession of Mercury's orbit in Newtonian gravity, gives a value of -0.6515572 arcseconds per century.

Though this is not the zero precession expected, there is little to correct this aside from reducing the time step. Extrapolation from higher values of k as done for the next case would just give a value of 0.

Using graph 11 to calculate the rate of precession of Mercury's orbit when the effect of general relativity is introduced, gives a value of 43.04907 arcseconds per century.

Though this is quite close to the true value, the presence of bands in the graph, even with it's general linear trend, led us to a different method for calculating the precession rate for this scenario.

By extrapolating the value of the rate of precession rates from values from orbits with higher ks, we obtained a range of 10 values. Together, they give us a rate of precession of 43.39841 ± 0.8049581 arcseconds per century.

Conclusion

The rate of precession of Mercury's orbit in Newtonian gravity is -0.6515572 arcseconds per century. Adding the correction term from General Relativity gives us a precession rate of 43.39841 ± 0.8049581 arcseconds per century.

The presence of numerical errors in our simulated data, is cause for interest. Though we are not yet sure where they come from, understanding their origins could be something more to work towards.

Finally, the extremely small time step we used might also be something to look into. Whether better results might be obtained from higher time steps by using different methods is also something to follow up on.

References

- Körber, C; Hammer, I; Wynen, J-L; Heuer, J; Müller, C; Hanhart, C (2018). A primer to numerical simulations: the perihelion motion of Mercury. Physics Education, 53(5), 055007–. doi:10.1088/1361-6552/aac487
- Phookun, Bikram. "Planetary orbits as simple harmonic motion." Resonance 8.12 (2003): 83-91. doi:10.1007/bf02839055.

Appendix

These are the Python notebooks we used for this project.

- **Main:** Calculating the Precession Rate
- **Supplementary:** Testing the Integration Algorithms